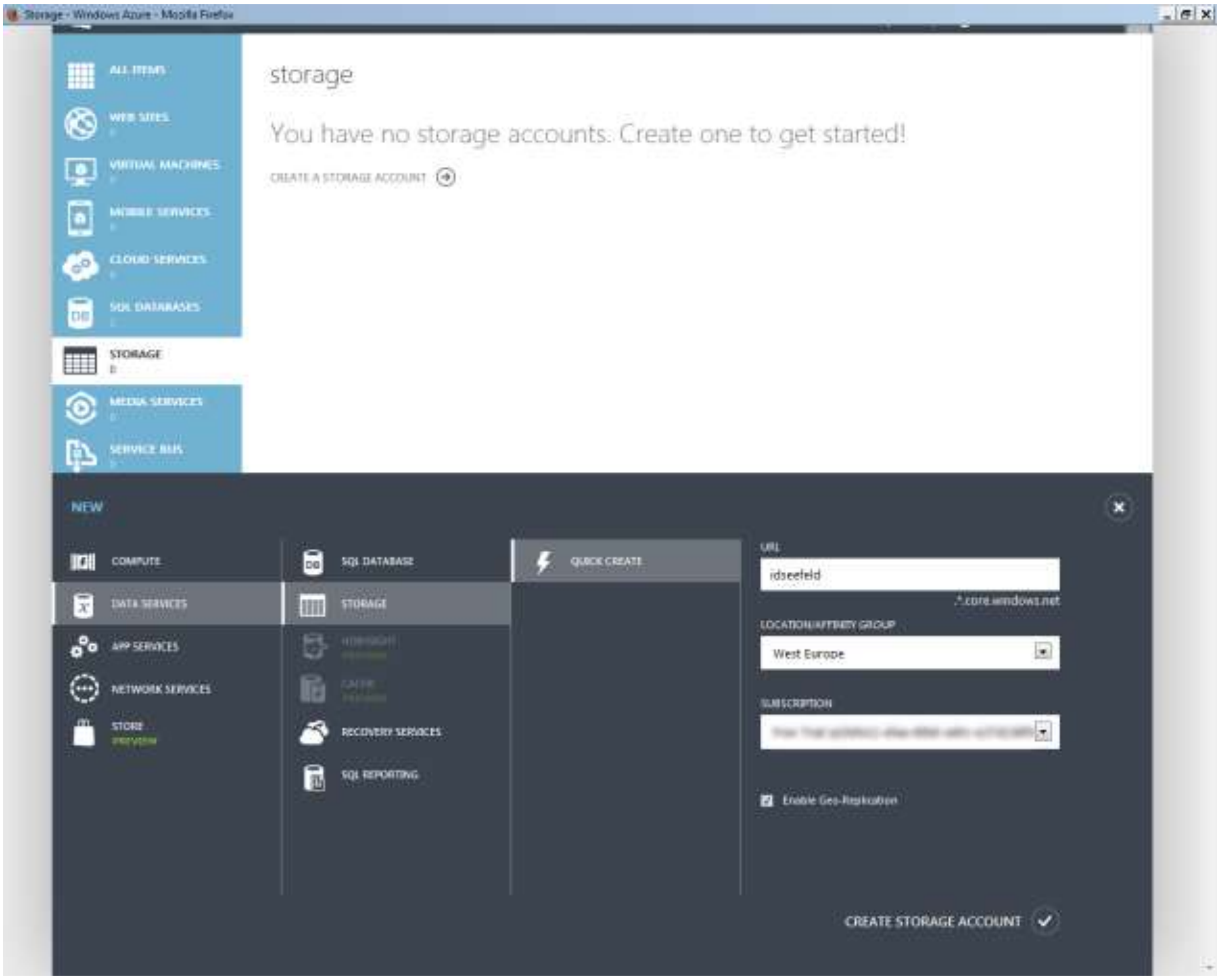
Azure Blob Storage Provider Documentation

What is the Azure Blob Storage Provider for Umbraco?

The Azure Blob Storage Provider replaces Umbracos default provider for media files. ***Full trust is required!***
The following steps show how to setup the *ABSP*:

1. At first you create the Azure storage.



storage

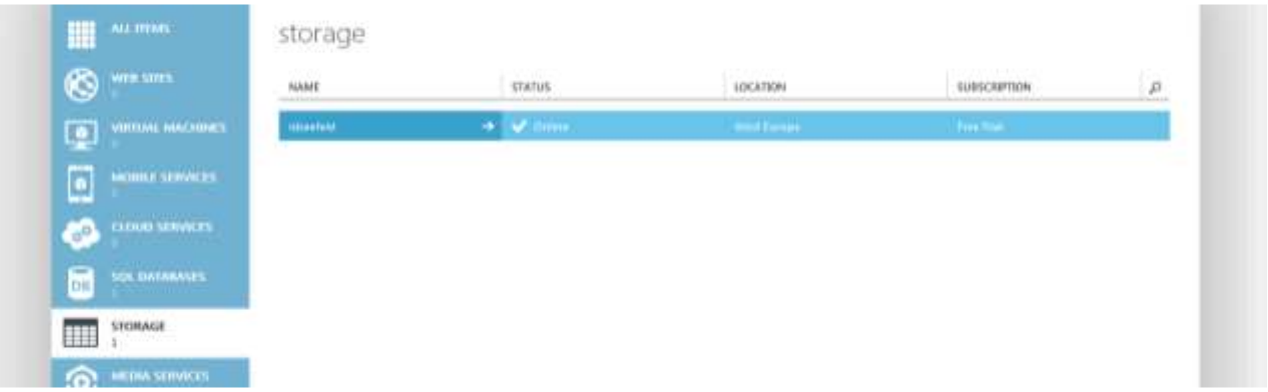
You have no storage accounts. Create one to get started!

URL: idseefeld .core.windows.net

LOCATION/AFFINITY GROUP: West Europe

☑ Enable Geo-Replication

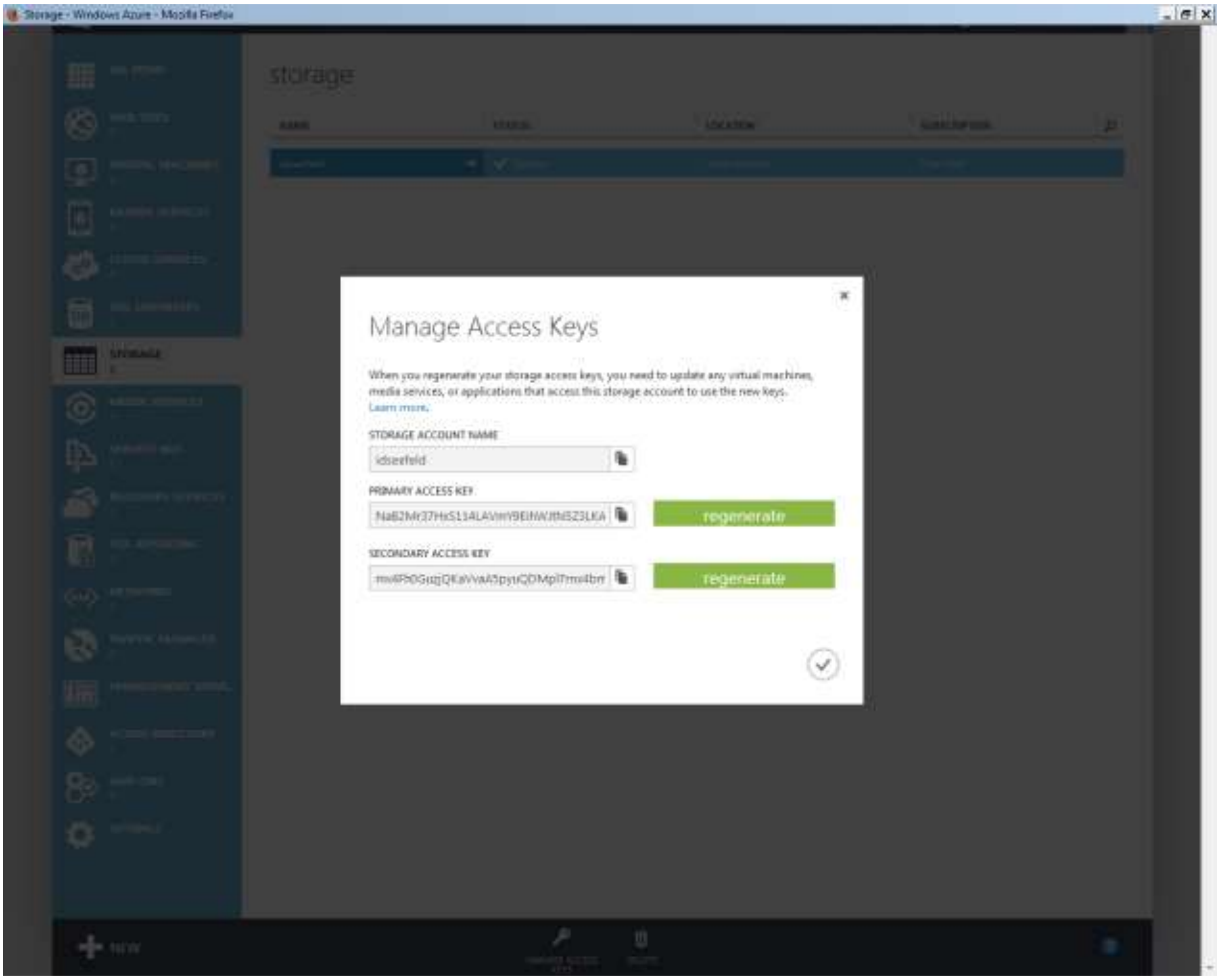
The Result



storage

NAME	STATUS	LOCATION	SUBSCRIPTION
idseefeld	✓ Online	West Europe	Free Trial

2. Copy one of the two generated keys.



Manage Access Keys

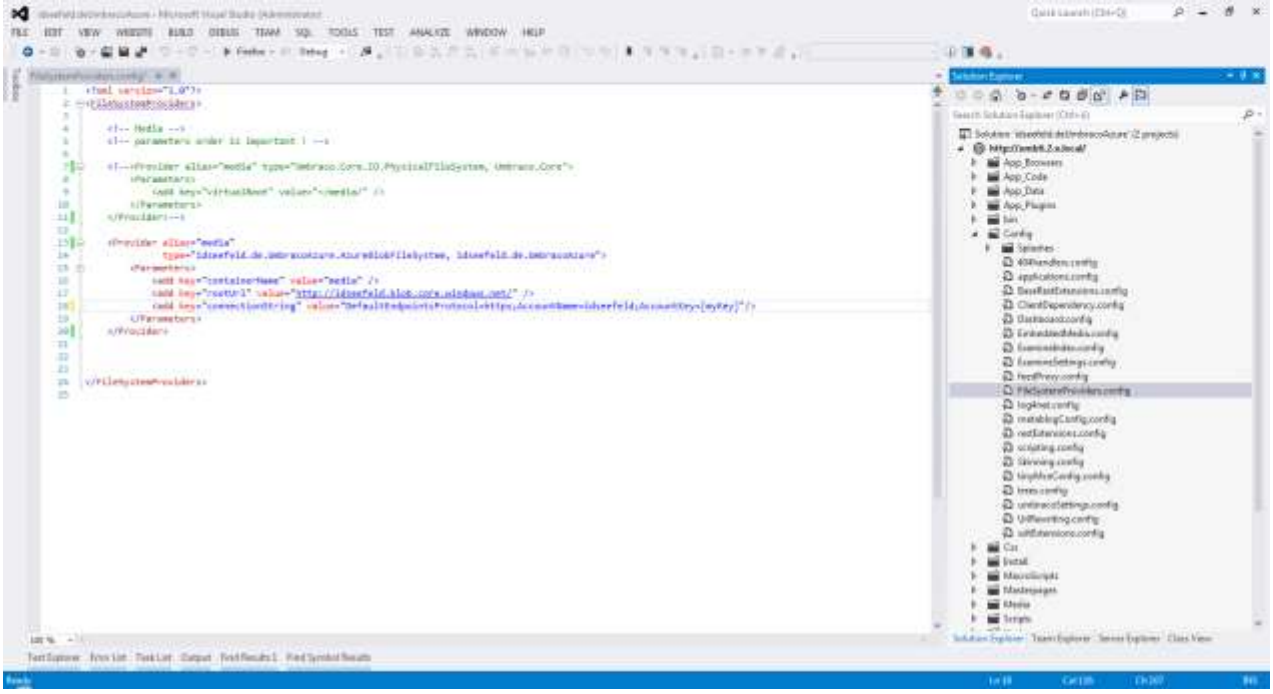
When you regenerate your storage access keys, you need to update any virtual machines, media services, or applications that access this storage account to use the new keys. Learn more.

STORAGE ACCOUNT NAME: idseefeld

PRIMARY ACCESS KEY: NaB2Mr37HxS114LAVmV9EINWJtN5Z3LKA

SECONDARY ACCESS KEY: mv4Pi0GuqjQKaVvaA5pyuQDMplTmx4bm

3. Open file ~/Config/FileSystemProviders.config of your Umbraco installation and paste your account name and key into the configuration:

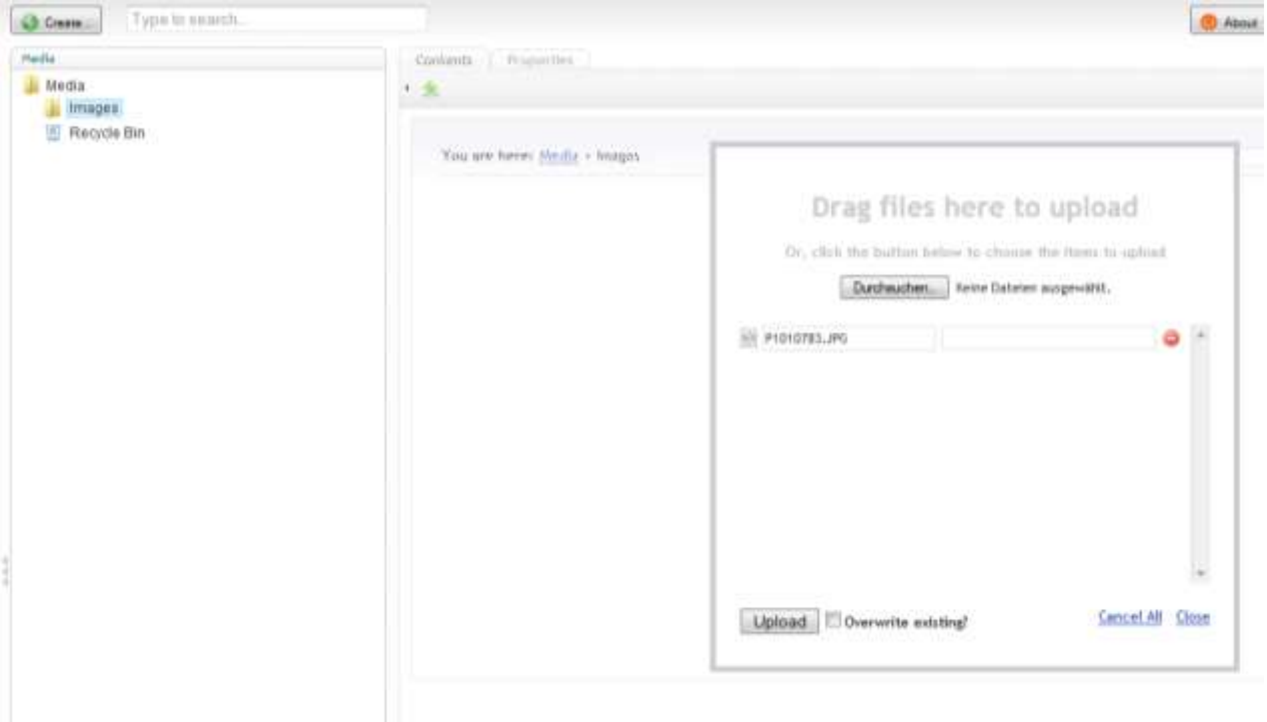


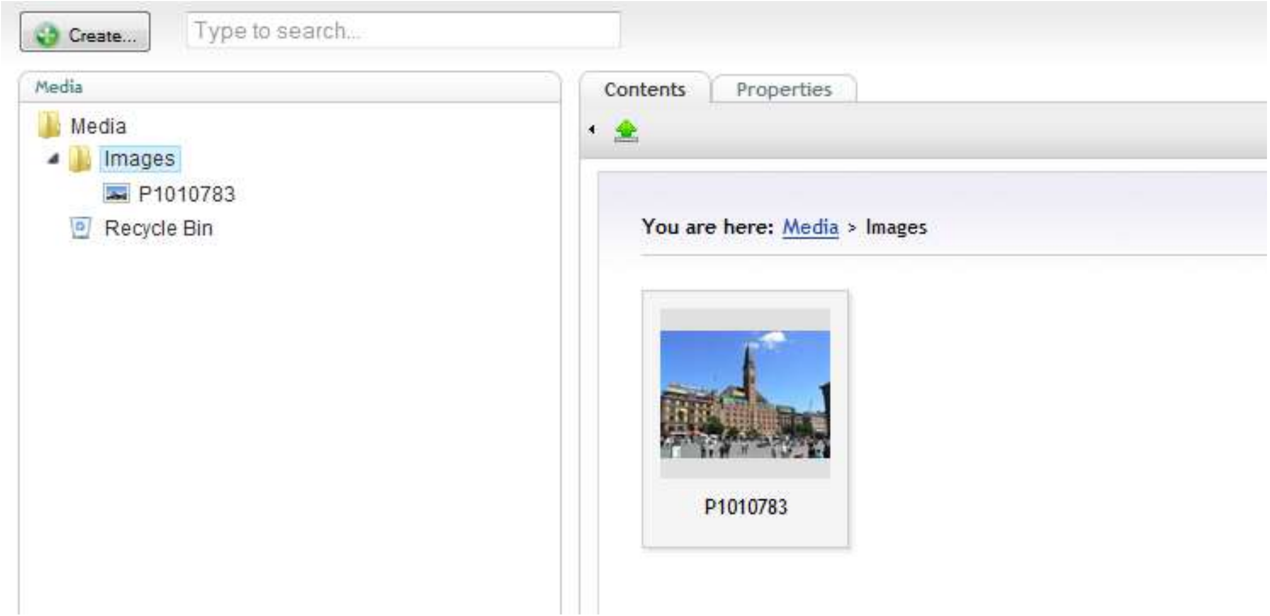
```
<?xml version="1.0"?>
<FileSystemProviders>

  <!-- Media -->
  <!-- parameters order is important ! -->

  <!--<Provider alias="media" type="Umbraco.Core.IO.PhysicalFileSystem, Umbraco.Core">
    <Parameters>
      <add key="virtualRoot" value="~/media/" />
    </Parameters>
  </Provider>-->

  <Provider alias="media"
            type="idseefeld.de.UmbracoAzure.AzureBlobFileSystem, idseefeld.de.UmbracoAzure">
    <Parameters>
      <add key="containerName" value="media" />
      <add key="rootUrl" value="http://idseefeld.blob.core.windows.net/" />
      <add key="connectionString" value="DefaultEndpointsProtocol=https;AccountName=idseefeld;AccountKey=[myKey]"/>
    </Parameters>
  </Provider>



</FileSystemProviders>
```

```xml
<?xml version="1.0"?>
<FileSystemProviders>
    <!-- Media -->
    <!-- parameters order is important ! -->

    <Provider alias="media"
                          type="idseefeld.de.UmbracoAzure.AzureBlobFileSystem,
idseefeld.de.UmbracoAzure">
                <Parameters>
                        <add key="containerName" value="media" />
                        <add key="rootUrl" value="http://[myAccountName].blob.core.windows.net/" />
                        <add key="connectionString"
value="DefaultEndpointsProtocol=https;AccountName=[myAccountName];AccountKey=[myAccountKey]"/>
                        <add key="mimetypes" value="woff|application/x-font-woff;mp3|audio/mpeg3" />
                        <add key="cacheControl" value="*|public, max-age=31536000;js|no-store" />
                </Parameters>
        </Provider>

</FileSystemProviders>
```

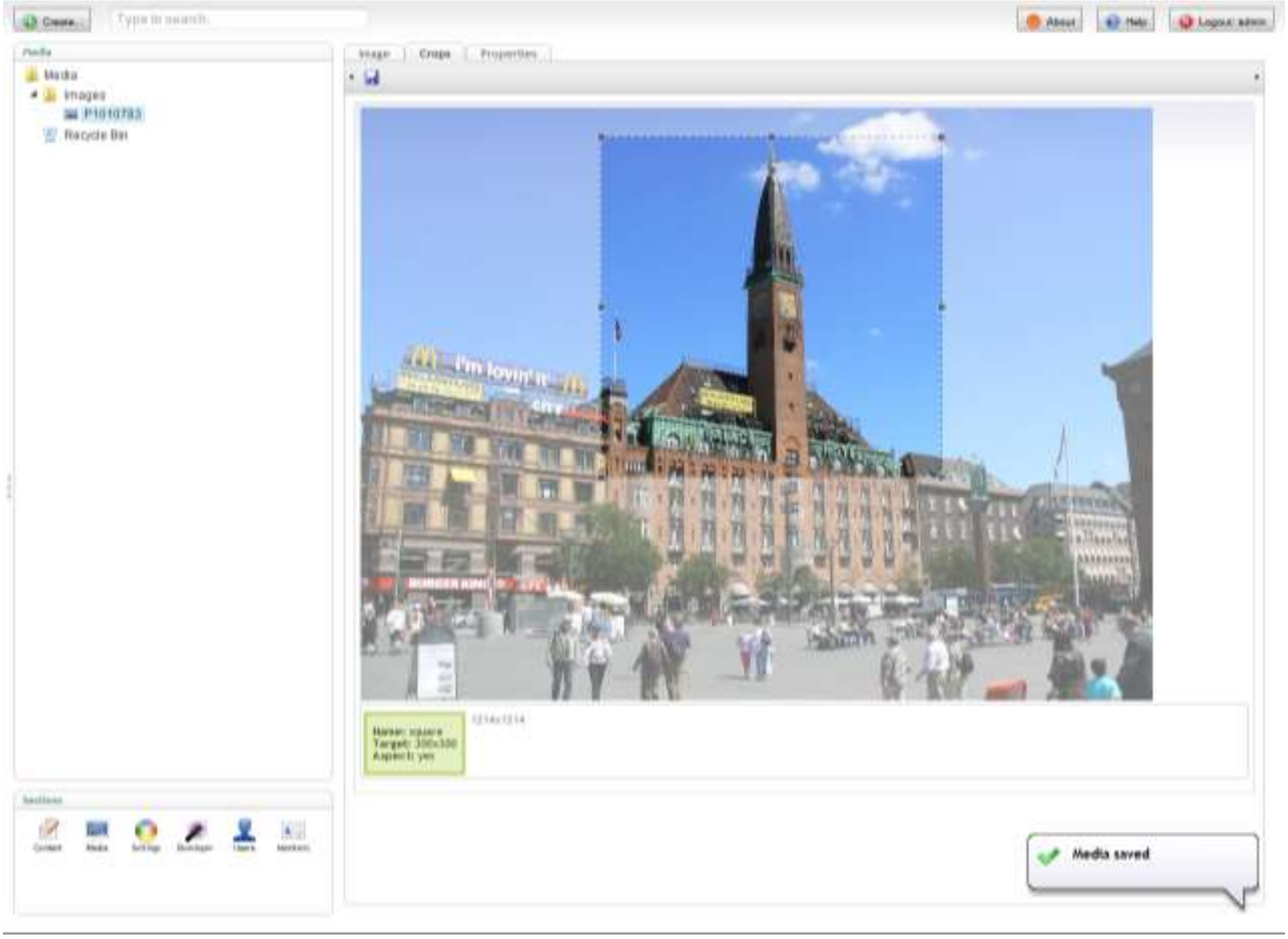
Since package version 1.0.7 you can add mimetypes. Separate each type description by ; (semicolon) and file extension (without leading dot) form header string by | (pipe). By default the mimetypes are taken from your website configuration. But you can override these if you like.
Cache-control can be configured since version 1.0.10. Add your settings to the parameters with key=”cacheControl”. Start with the wildcard (*) setting and separate the file type extension by | (pipe) from the cache-control settings. Separate each file type | cache-control group by ; (semicolon).
Allowed cache-controls are:
`public` *or* `public, max-age=12345678` (*or simply:* `max-age=12345678`)
`private` *or* `private, max-age=12345678`
`no-store`
`no-cache`
The max-age value is defined in seconds - i.e. 31536000 means one year (60 x 60 x 24 x 365).

4. Restart the web server (e. g. touch web.config) and upload an image.

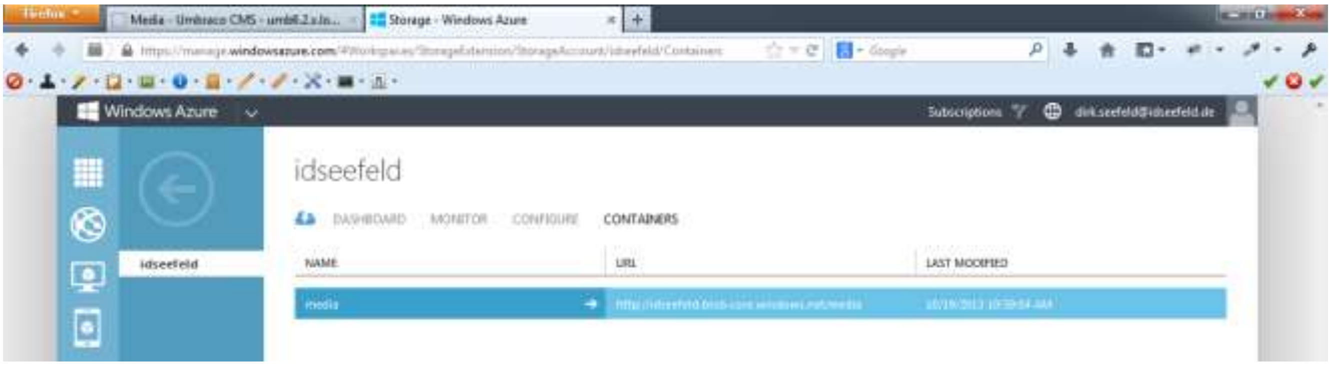




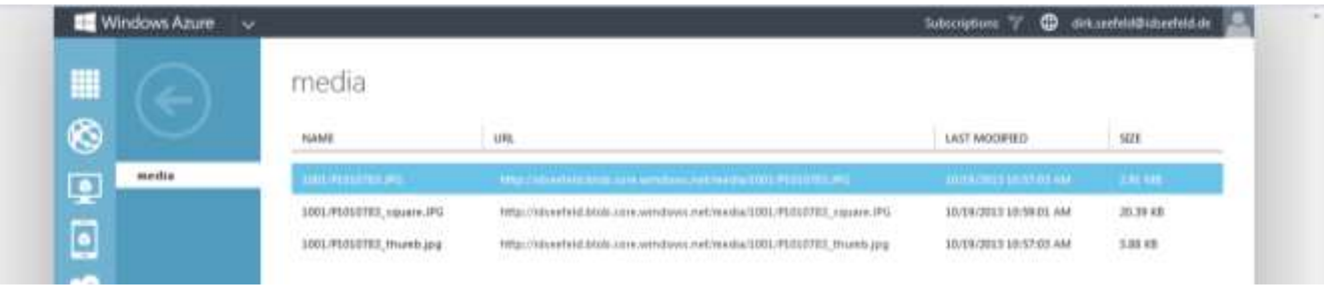
In my example I have previously installed the Image Cropper Extended and defined crops for media and content (for details see Image Cropper Extended documentation). *Just to show that it works!*



5. Check that your uploaded images are in the cloud ...

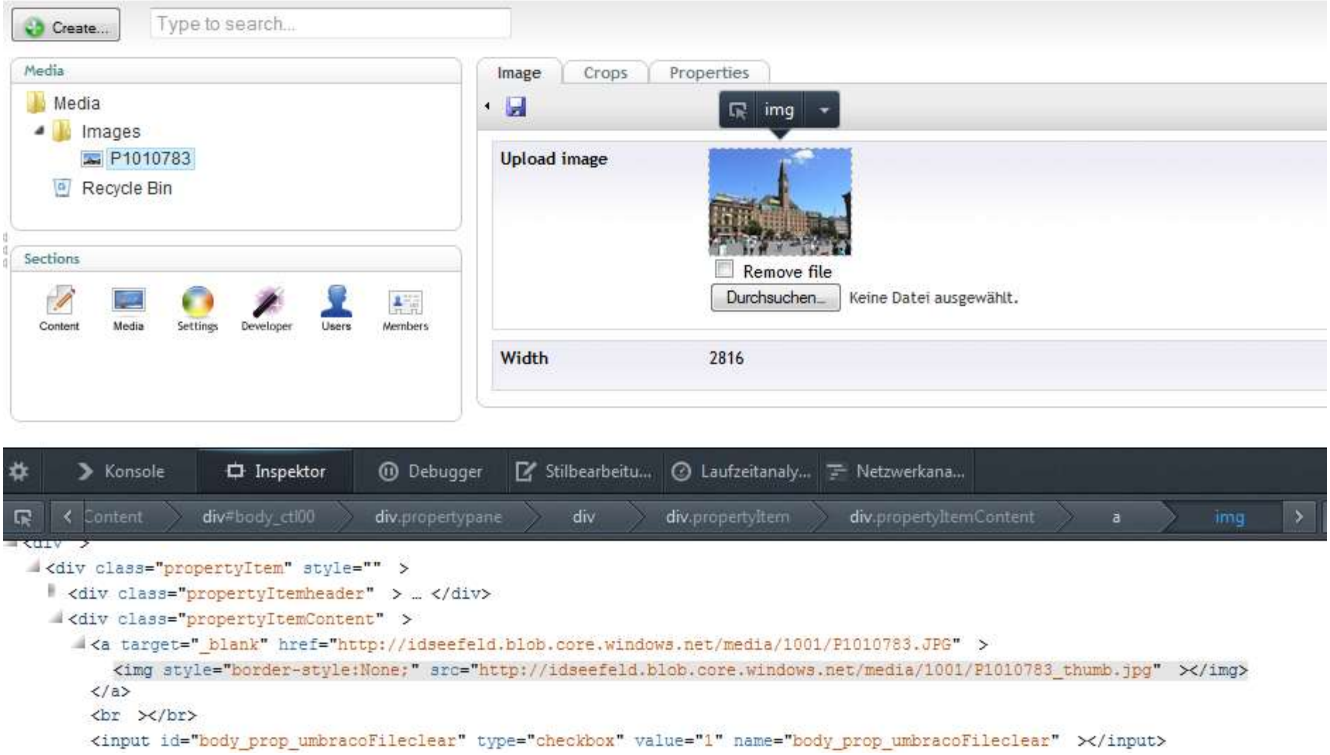


NAME	URL	LAST MODIFIED
media	http://idseefeld.blob.core.windows.net/media	10/19/2013 10:39:04 AM



NAME	URL	LAST MODIFIED	SIZE
1001/P1010783.JPG	http://idseefeld.blob.core.windows.net/media/1001/P1010783.JPG	10/19/2013 10:57:03 AM	2.81 MB
1001/P1010783_square.JPG	http://idseefeld.blob.core.windows.net/media/1001/P1010783_square.JPG	10/19/2013 10:59:01 AM	20.39 KB
1001/P1010783_thumb.jpg	http://idseefeld.blob.core.windows.net/media/1001/P1010783_thumb.jpg	10/19/2013 10:57:03 AM	5.88 KB

6. ... or explore the html.



Upload image

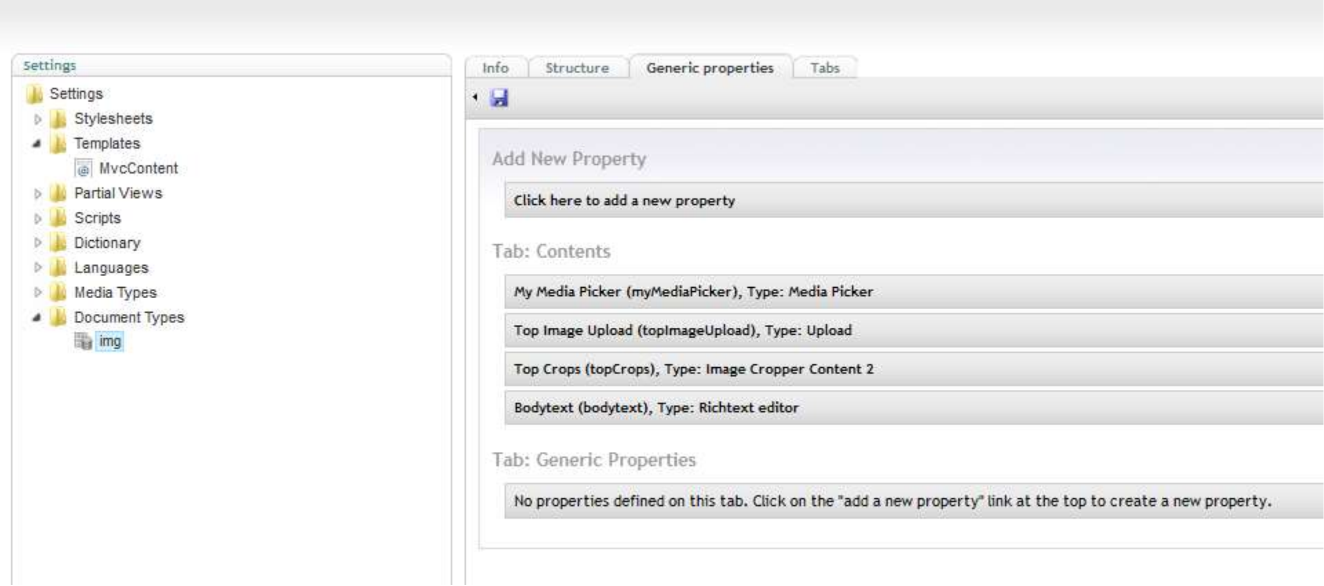
☐ Remove file

Durchsuchen... Keine Datei ausgewählt.

Width 2816

```
<div class="propertyItem" style="" >
 <div class="propertyItemheader" > … </div>
 <div class="propertyItemContent" >
  <a target="_blank" href="http://idseefeld.blob.core.windows.net/media/1001/P1010783.JPG" >
   <img style="border-style:None;" src="http://idseefeld.blob.core.windows.net/media/1001/P1010783_thumb.jpg" ></img>
  </a>
  <br ></br>
  <input id="body_prop_umbracoFileclear" type="checkbox" value="1" name="body_prop_umbracoFileclear" ></input>
```

7. Example of a document type



Add New Property

Click here to add a new property

Tab: Contents

My Media Picker (myMediaPicker), Type: Media Picker

Top Image Upload (topImageUpload), Type: Upload

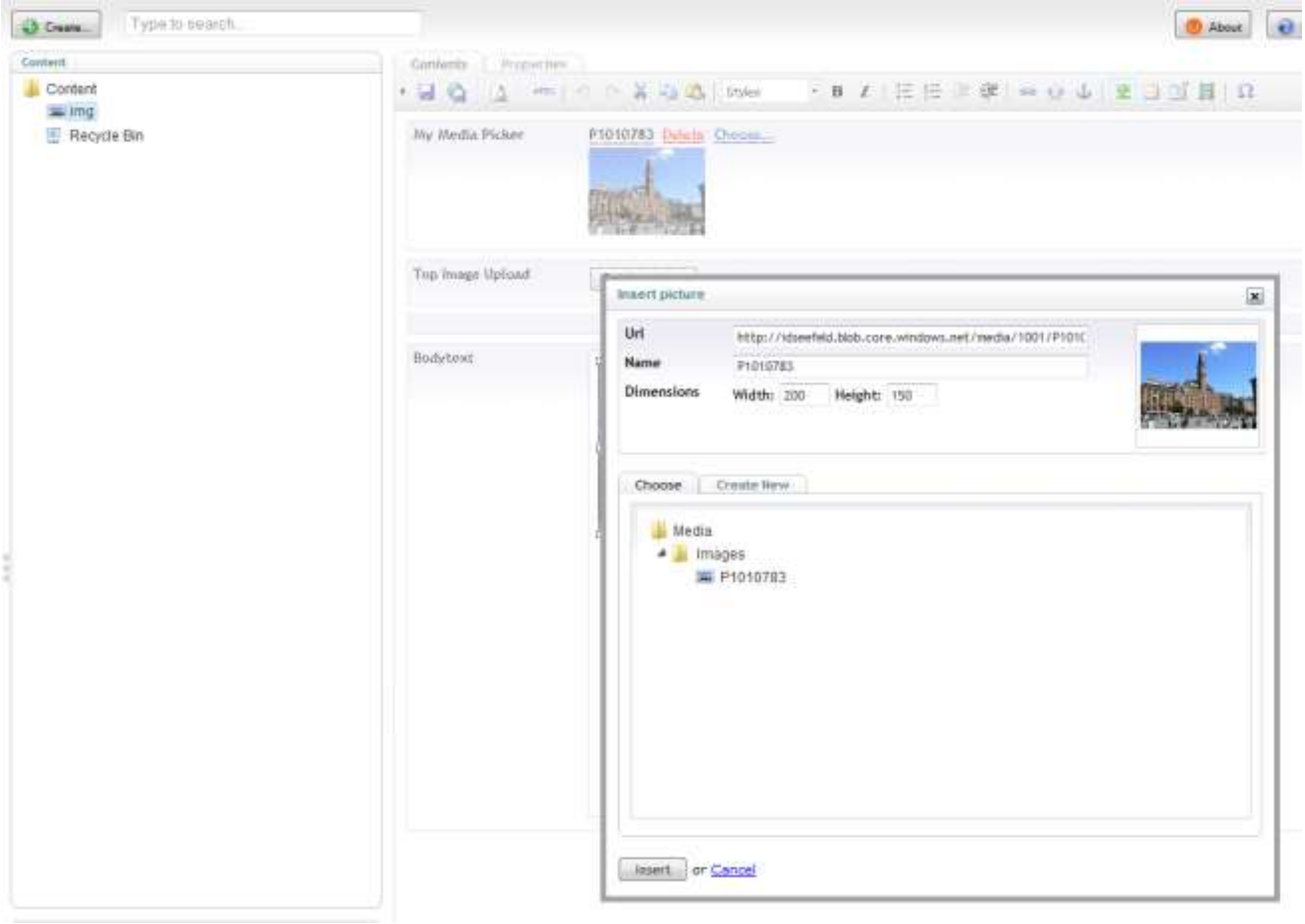
Top Crops (topCrops), Type: Image Cropper Content 2

Bodytext (bodytext), Type: Richtext editor

Tab: Generic Properties

No properties defined on this tab. Click on the "add a new property" link at the top to create a new property.

8. Create a node of the *img* document type and choose an image with the media Picker and insert one into the rich text editor.



My Media Picker: P1010783 Delete Choose...

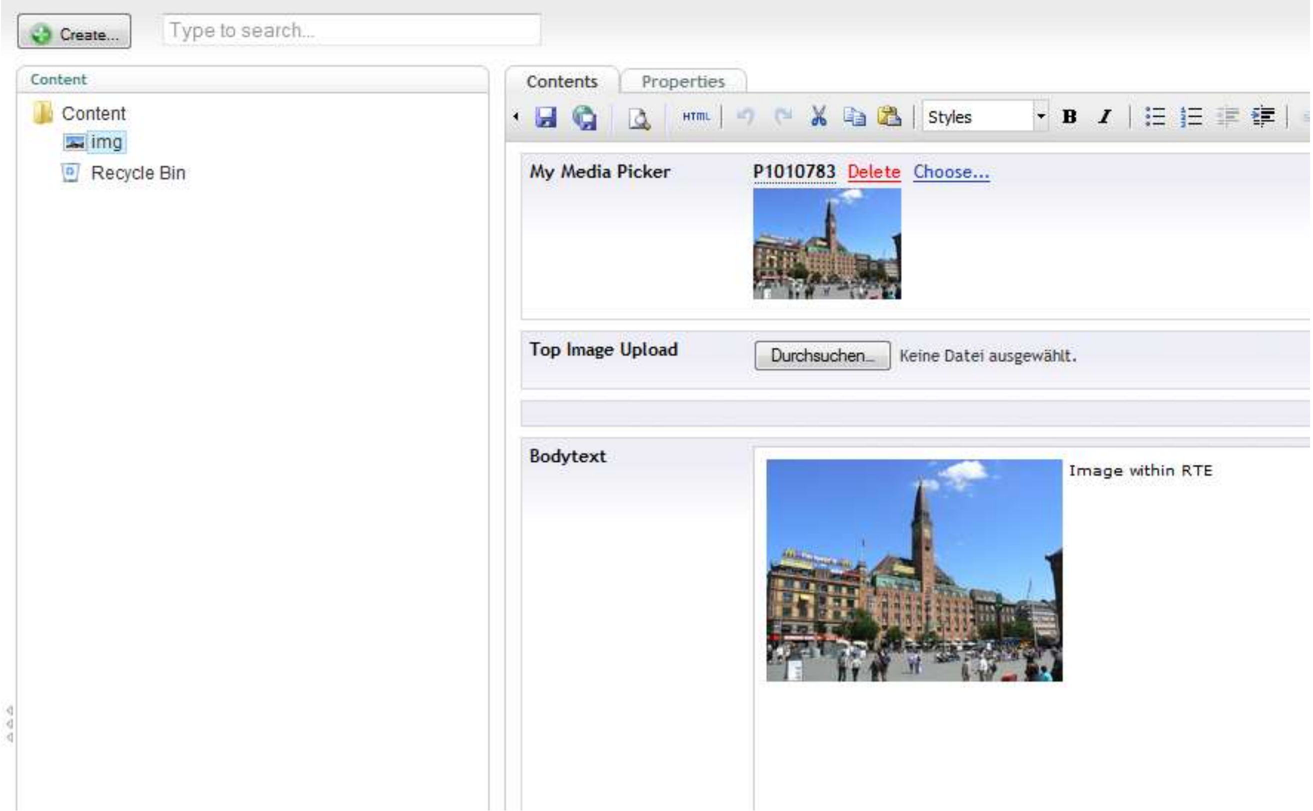
Top Image Upload

Bodytext

Url: http://idseefeld.blob.core.windows.net/media/1001/P101C

Name: P1010783

Dimensions: Width: 200 Height: 150

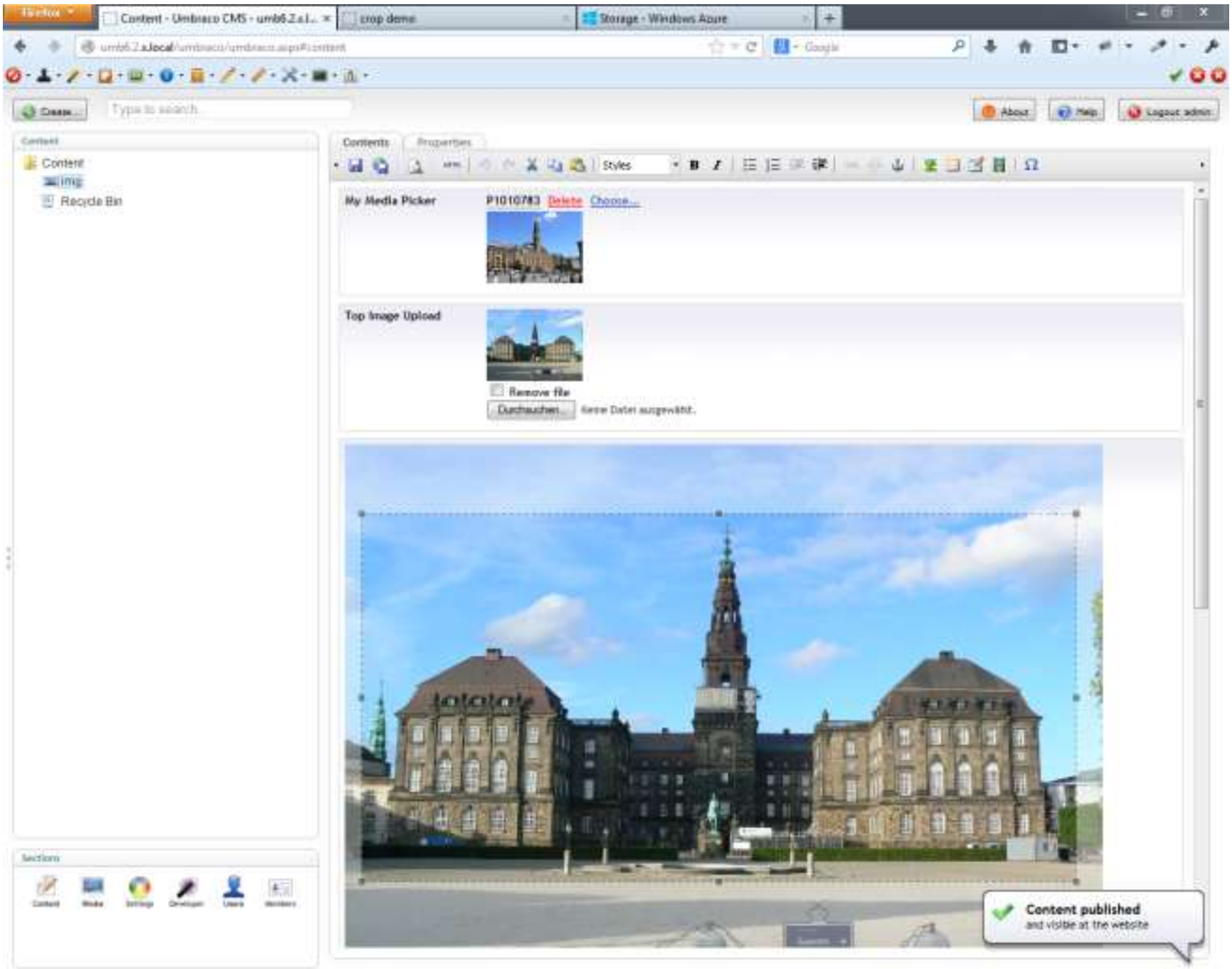


My Media Picker: P1010783 Delete Choose...

Top Image Upload: Durchsuchen... Keine Datei ausgewählt.

Bodytext: Image within RTE

9. Upload another image with the upload, adjust the associated cropper and save & publish the node.

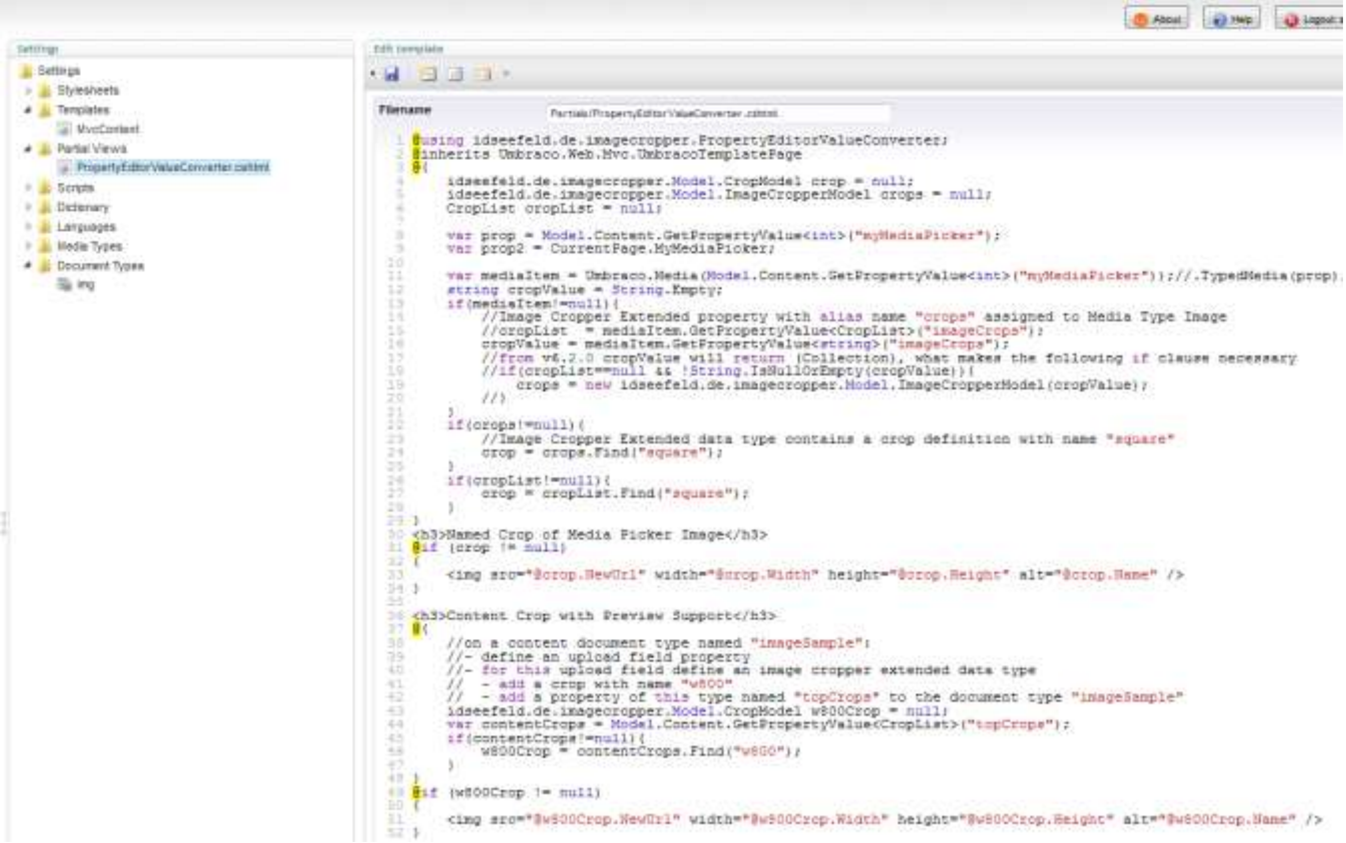


My Media Picker: P1010783 Delete Choose...

Top Image Upload: □ Remove file; Durchsuchen... Keine Datei ausgewählt.

Content published
and visible at the website

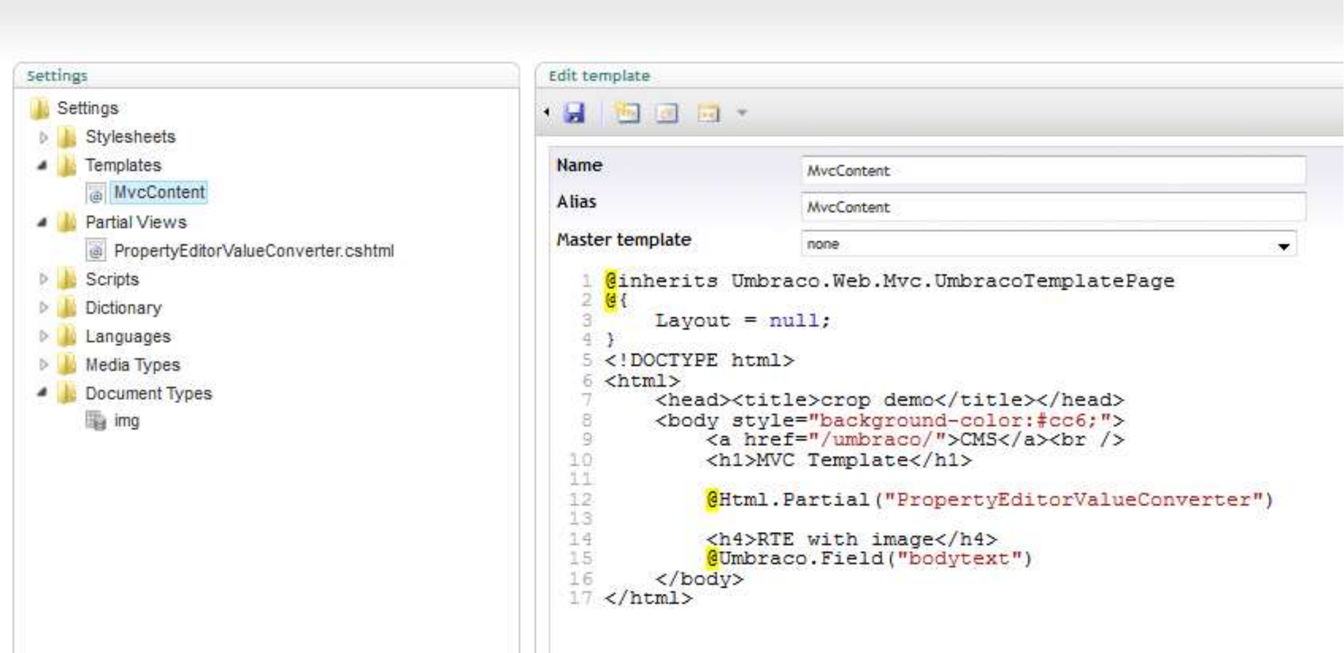
10. Sample scripts for a partial view ...



Filename: Partials/PropertyEditorValueConverter.cshtml

```
@using idseefeld.de.imagecropper.PropertyEditorValueConverter;
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    idseefeld.de.imagecropper.Model.CropModel crop = null;
    idseefeld.de.imagecropper.Model.ImageCropperModel crops = null;
    CropList cropList = null;

    var prop = Model.Content.GetPropertyValue<int>("myMediaPicker");
    var prop2 = CurrentPage.MyMediaPicker;

    var mediaItem = Umbraco.Media(Model.Content.GetPropertyValue<int>("myMediaPicker"));//.TypedMedia(prop).
    string cropValue = String.Empty;
    if(mediaItem!=null){
        //Image Cropper Extended property with alias name "crops" assigned to Media Type Image
        //cropList  = mediaItem.GetPropertyValue<CropList>("imageCrops");
        cropValue = mediaItem.GetPropertyValue<string>("imageCrops");
        //from v6.2.0 cropValue will return (Collection), what makes the following if clause necessary
        //if(cropList==null && !String.IsNullOrEmpty(cropValue)){
            crops = new idseefeld.de.imagecropper.Model.ImageCropperModel(cropValue);
        //}
    }
    if(crops!=null){
        //Image Cropper Extended data type contains a crop definition with name "square"
        crop = crops.Find("square");
    }
    if(cropList!=null){
        crop = cropList.Find("square");
    }
}
<h3>Named Crop of Media Picker Image</h3>
@if (crop != null)
{
    <img src="@crop.NewUrl" width="@crop.Width" height="@crop.Height" alt="@crop.Name" />
}

<h3>Content Crop with Preview Support</h3>
@{
    //on a content document type named "imageSample":
    //- define an upload field property
    //- for this upload field define an image cropper extended data type
    //  - add a crop with name "w800"
    //  - add a property of this type named "topCrops" to the document type "imageSample"
    idseefeld.de.imagecropper.Model.CropModel w800Crop = null;
    var contentCrops = Model.Content.GetPropertyValue<CropList>("topCrops");
    if(contentCrops!=null){
        w800Crop = contentCrops.Find("w800");
    }
}
@if (w800Crop != null)
{
    <img src="@w800Crop.NewUrl" width="@w800Crop.Width" height="@w800Crop.Height" alt="@w800Crop.Name" />
}
```

... and one for the MVC template, available in the documentation section of the package download page.



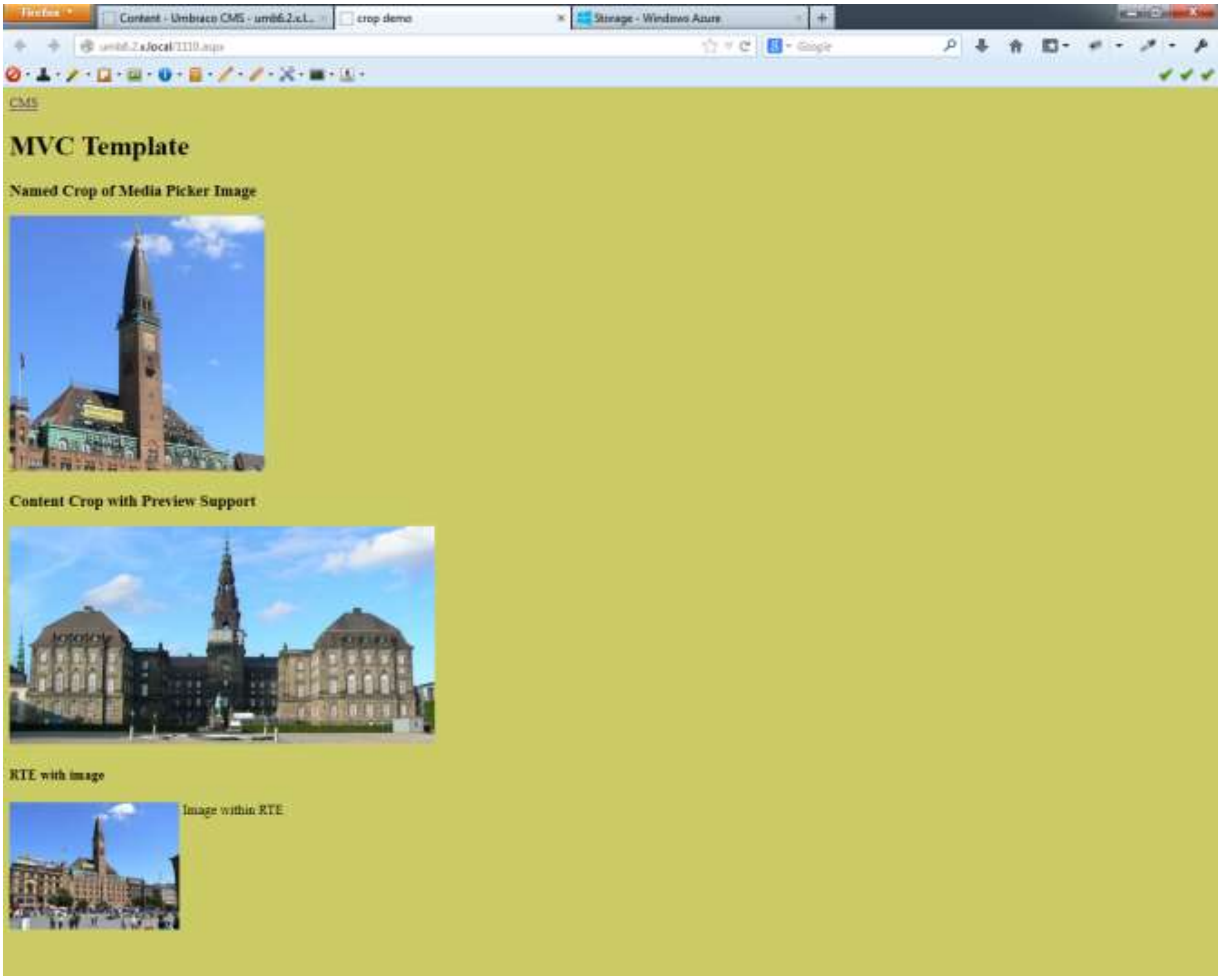
Name: MvcContent

Alias: MvcContent

Master template: none

```
@inherits Umbraco.Web.Mvc.UmbracoTemplatePage
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
    <head><title>crop demo</title></head>
    <body style="background-color:#cc6;">
        <a href="/umbraco/">CMS</a><br />
        <h1>MVC Template</h1>

        @Html.Partial("PropertyEditorValueConverter")

        <h4>RTE with image</h4>
        @Umbraco.Field("bodytext")
    </body>
</html>
```

11. Result view of the sample template.



CMS

MVC Template

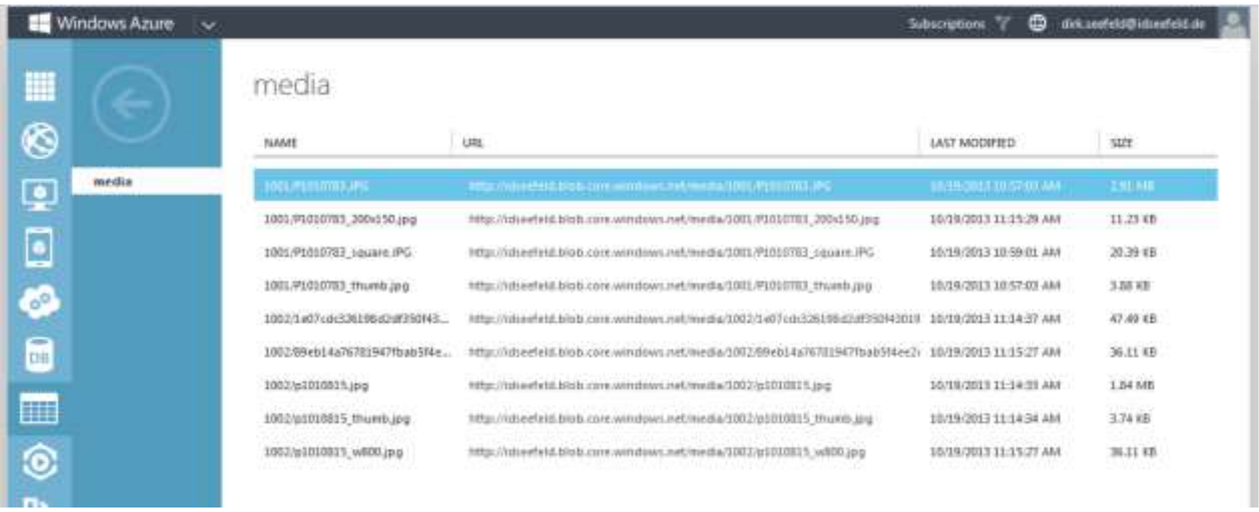
Named Crop of Media Picker Image

Content Crop with Preview Support

RTE with image

Image within RTE

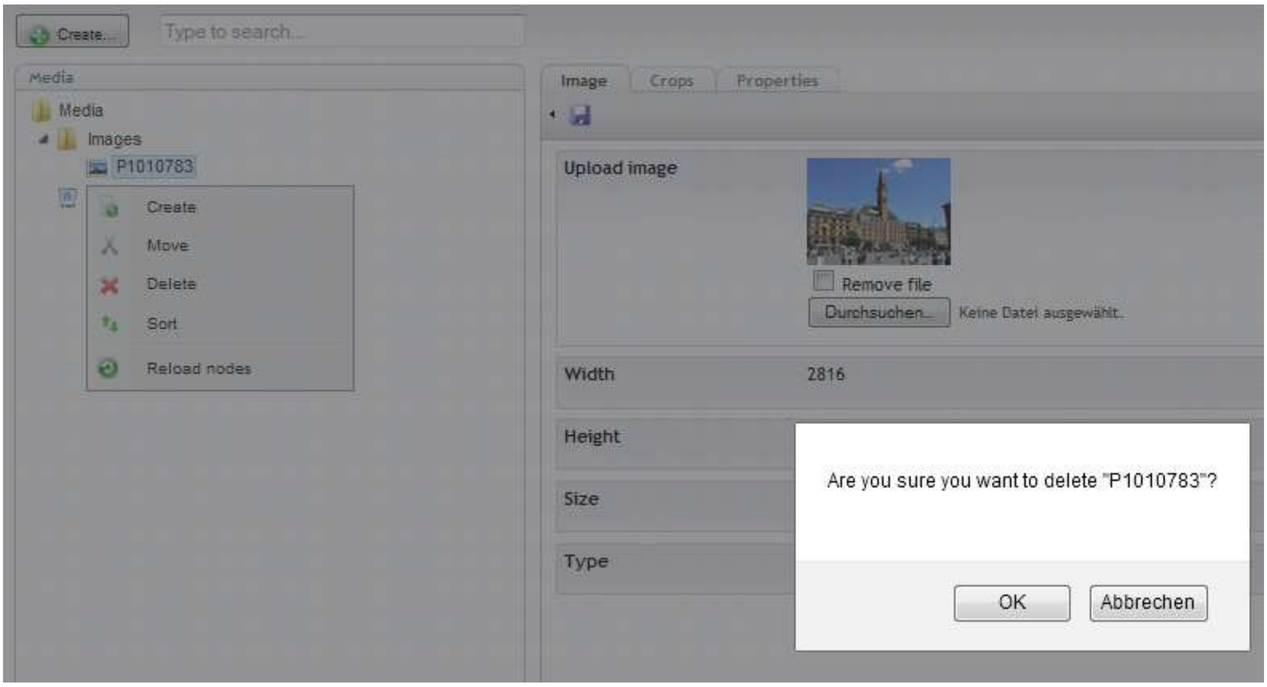
12. List of uploaded files.



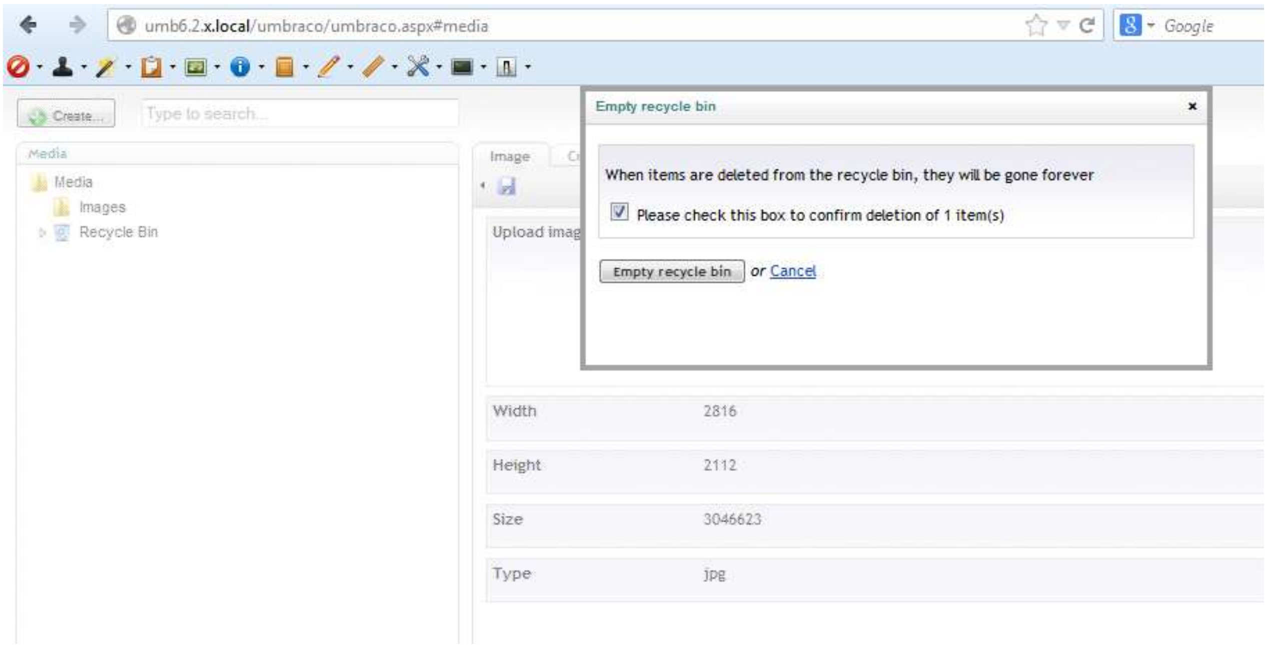
media

NAME	URL	LAST MODIFIED	SIZE
1001/P1010783.JPG	http://idseefeld.blob.core.windows.net/media/1001/P1010783.JPG	10/19/2013 10:57:03 AM	1.91 MB
1001/P1010783_200x150.jpg	http://idseefeld.blob.core.windows.net/media/1001/P1010783_200x150.jpg	10/19/2013 11:15:29 AM	11.23 KB
1001/P1010783_square.JPG	http://idseefeld.blob.core.windows.net/media/1001/P1010783_square.JPG	10/19/2013 10:59:01 AM	20.39 KB
1001/P1010783_thumb.jpg	http://idseefeld.blob.core.windows.net/media/1001/P1010783_thumb.jpg	10/19/2013 10:57:03 AM	3.88 KB
1002/1a07cdc326198d2df350f43...	http://idseefeld.blob.core.windows.net/media/1002/1a07cdc326198d2df350f43019	10/19/2013 11:14:37 AM	47.49 KB
1002/89eb14a76781947fbab5f4e...	http://idseefeld.blob.core.windows.net/media/1002/89eb14a76781947fbab5f4ee2	10/19/2013 11:15:27 AM	36.11 KB
1002/p1010815.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815.jpg	10/19/2013 11:14:33 AM	1.84 MB
1002/p1010815_thumb.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815_thumb.jpg	10/19/2013 11:14:34 AM	3.74 KB
1002/p1010815_w800.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815_w800.jpg	10/19/2013 11:15:27 AM	36.11 KB

13. Delete a media item.



Are you sure you want to delete "P1010783"?



Empty recycle bin

When items are deleted from the recycle bin, they will be gone forever

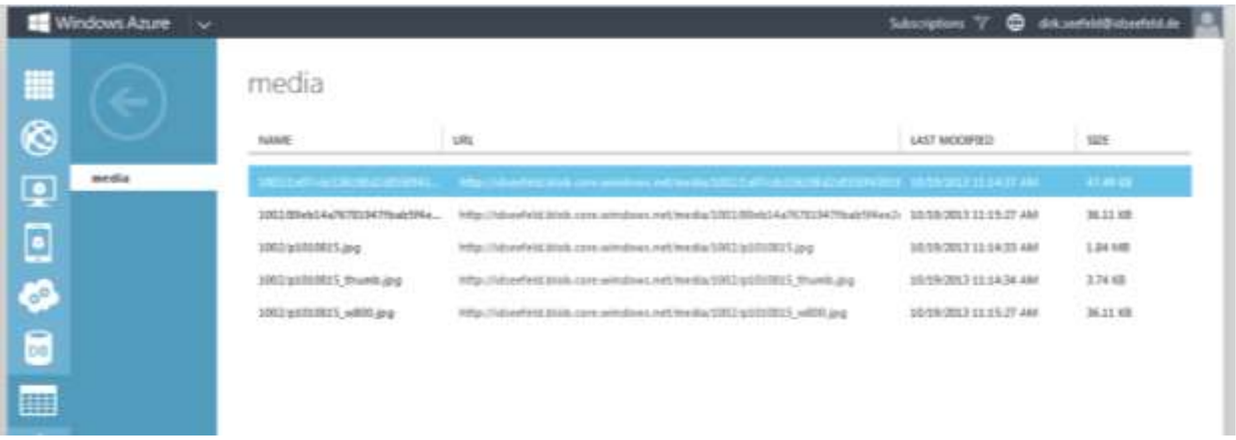
☑ Please check this box to confirm deletion of 1 item(s)

Empty recycle bin or Cancel

Width	2816
Height	2112
Size	3046623
Type	jpg

At the time of this writing (previous to version 6.2.0) there is an issue with deleting a single media item in the recycle bin.

But you can empty the whole recycle bin and all trashed items will be deleted in Azure storage.



media

NAME	URL	LAST MODIFIED	SIZE
[illegible]	[illegible]	10/19/2013 11:14:37 AM	47.49 KB
1001/00eb14a767019479bab5f4e...	http://idseefeld.blob.core.windows.net/media/1001/00eb14a767019479bab5f4ea2	10/18/2013 11:15:27 AM	36.11 KB
1002/p1010815.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815.jpg	10/19/2013 11:14:33 AM	1.84 MB
1002/p1010815_thumb.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815_thumb.jpg	10/19/2013 11:14:34 AM	3.74 KB
1002/p1010815_w800.jpg	http://idseefeld.blob.core.windows.net/media/1002/p1010815_w800.jpg	10/19/2013 11:15:27 AM	36.11 KB

An issue report with a possible solution is created, but I don't know whether it will make it into the release version 6.2.0 - your vote might help.

14. Bug Fix

Until version 1.0.5 there was a bug that lead to doubled directory names (actually numbers starting with 1001) for different uploads. In case of deleting an item this actually deletes the whole directory, as this is assumed unique for only one item and siblings like thumbnails for images.

Version 1.0.8 writes a control file to the repository with the number of the latest existing media directory in case you have data from a previous version. Otherwise, it contains 0. This prevents “old” incorrectly named blobs from deletion. Of course, this is only a work-a-round. However, it is the best solution I could figure out so far.